

Controller behaviour-based e Ibrido per
“Surface and Perimeter Coverage” e
“Shelter with constraint Access”

Progetto Sistemi Intelligenti Robotici

Andrea Lanari

Settembre 2020

1 Introduzione

L'idea alla base del progetto è quella di riproporre due degli esperimenti affrontati in [1] facendo uso di controller behaviour based e ibridi.

Nello specifico, per quanto riguarda le architetture behaviour based, verranno utilizzati i modelli a Motor Schema e quelli a Sussunzione.

L'obiettivo finale è quindi quello di valutare i vari controller attraverso l'attribuzione di un punteggio ad esperimenti simulati: il "Surface and Perimeter Coverage" (SPC) e il "Shelter with Constraint Access" (SCA).

Per fare questo, si è fatto uso di ARGoS [2], un software open source che fornisce un ambiente di simulazione per la swarm robotics.

2 Protocollo degli esperimenti

Al fine di proporre un'adeguata comparazione tra i vari modelli di controller, sono stati ripresi i protocolli descritti in [1], di seguito riportati:

- **Protocollo generale:** Gli esperimenti sono stati effettuati simulando 20 robots E-puck (disponibili in ARGoS) posti in un'arena dodecagonale di area 4.91 m^2 circondata da muri; il pavimento è di colore grigio e il tempo dato per raggiungere l'obiettivo è $T = 120 \text{ s}$.

In particolare, gli E-puck sono dotati dei seguenti attuatori/sensori:

- un sensore che riconosce il colore del pavimento (ne sono effettivamente disponibili 3 ma in questo progetto soltanto uno è stato utilizzato, poiché si è voluto riprendere quanto fatto negli esperimenti originali);
- 8 sensori di luce e di prossimità disposti lungo la circonferenza del robot;
- un dispositivo "Range and Bearing", che ha funzionalità sia da attuatore che da sensore, che permette lo scambio di 10 Byte di dati oltre a fornire la direzione e la distanza del robot con cui si sta comunicando;
- un motore che permette il movimento del robot ad una velocità massima di 16 cm s^{-1} .

- **Regole aggiuntive per SCA:** per questo esperimento, è stata aggiunta all'arena una patch bianca rettangolare ($0.6 \times 0.15 \text{ m}$) posta al centro e circondata da muri sui lati nord, ovest ed est di dimensione ($L \times 0.05 \times 0.05$) con L che assume valori 0.15 e 0.6.

La funzione obiettivo è calcolata 10 volte al secondo, utilizzando questa formula:

$$F_{SCA} = \sum_{t=1}^T N(t)$$

dove $N(t)$ è il numero di robot N al tempo t presenti all'interno del riparo. Quindi, al termine dell'esperimento, più il punteggio sarà alto, maggiore sarà la valutazione del controller.

- **Regole aggiuntive per SPC:** nell'esperimento vengono usate due patch, una bianca rettangolare di lato 0.6 m e una nera circolare di raggio 0.6 m. La funzione obbiettivo, calcolata al termine dell'esperimento è la seguente:

$$F_{CFA} = E[d_a(T)]/c_a + E[d_p(T)]/c_p$$

dove $E[d_a(T)]$ è la distanza al tempo finale T , tra un generico punto nel quadrato e il più vicino robot all'interno della regione, mentre $E[d_p(T)]$ è la distanza tra un generico punto nel perimetro del cerchio e il più vicino robot sulla circonferenza. $c_a = 0.08$ e $c_p = 0.06$ sono i valori di scaling che rappresentano la condizione ideale in cui 9 robot sono regolarmente ed equamente distribuiti lungo il perimetro e l'area. A differenza della SCA, al miglior esito corrisponde il punteggio più basso.

3 I controller

Come citato nell'introduzione, questo progetto mira a comparare tre differenti tipi di architettura: MotorSchema, Sussunzione e Ibrido.

Il Motorschema è un modello di computazione concorrente e distribuito appartenente alla famiglia delle architetture Behaviour-based. Esso è solitamente organizzato in Schemas, che forniscono un insieme di comportamenti primitivi, sui quali è possibile costruirne di più complessi.

In particolare, alla base di questo modello ci sono i perceptual schemas, che forniscono informazioni dall'ambiente ad un determinato comportamento, il quale elabora l'input ricevuto creando un vettore azione.

Questo output viene quindi sommato a tutti gli altri vettori azione dei vari comportamenti descritti nel controller, al fine di ottenere un vettore azione finale, il quale viene successivamente fornito all'attuatore (nel nostro caso esso necessita prima di essere convertito in "wheels-speed").

Per quanto riguarda questo progetto, si è implementato un motorschema per ogni sensore utilizzato: proximitySchema, friendSchema (legato al Range and Bearing), lightSchema (solo in SCA) e il RandomSchema. Quest'ultimo non è legato direttamente ad un sensore, ma risulta necessario per muovere il robot casualmente alla ricerca di qualche stimolo ambientale, nel momento in cui non ne ha.

Anche il modello a Sussunzione appartiene alla famiglia Behaviour-based, ma a differenza del MotorSchema decompone l'architettura del robot in task-achieving behaviour o competenze, applicando un controllo parallelo e distribuito.

L'idea alla base della Sussunzione è quella di costruire un livello di controllo per

ogni competenza andando ad aggiungerne di nuovi nel momento in cui si decide di aggiungere una nuovo task. Questa design incrementale prevede che i livelli più alti siano a conoscenza dell'esistenza di quelli più bassi e possano sfruttarli, attivandoli o inibendoli, per raggiungere il proprio goal. Sostanzialmente, i livelli alti sussumono il ruolo di quelli più bassi quando essi vorrebbero prendere il controllo. Dunque, a livello implementativo, è possibile organizzare questo tipo di architettura in vari modi: come una macchina a stati finiti aumentata, come una cascata di if...then..else o con alberi comportamentali. In questo progetto è stata scelta quest'ultima implementazione poiché struttura molto reattiva e modulare. In particolar modo, si è fatto uso della libreria esterna Luabt [3], che permette la creazione di alberi comportamentali in maniera rapida e coincisa. Al fine di creare una struttura che rispecchi perfettamente l'architettura a sussunzione, è necessario però rispettare alcune regole:

- costruire la cascade dei comportamenti ordinandoli dalla priorità più alta a quella più bassa;
- non far mai ritornare il Success ma solo la Failure o il Running;
- ogni comportamento deve contenere la propria condizione di attivazione e il proprio codice di esecuzione.

Anche in questo caso, i vari comportamenti sono tutti legati ai vari sensori come nel caso del MotorSchema.

L'ultimo approccio, quello ibrido, prevede l'abbinamento dei comportamenti ad un'architettura deliberativa.

In questo progetto, si è deciso di sviluppare questo tipo di architettura riprendendo il classico ciclo del deliberativo sense-plan-act, implementando la parte di pianificazione attraverso comportamenti attivati da funzioni che elaboravano la parte sensoriale e che si concludevano azionando gli attuatori.

Come per gli altri due tipi di controller, anche in questo caso, i vari behaviour sono stati costruiti sulla base dei sensori utilizzati durante l'esperimento.

Fin qui si è descritto quali tipologie di controller sono state utilizzate, senza specificare i vari comportamenti che sono stati effettivamente implementati.

Per quanto riguarda la SPC, si è deciso di far muovere inizialmente i robot in maniera randomica, con collision avoid per rendere più fluido l'allontanamento dai muri. Nel momento in cui un robot giunge in una delle due patch, setta l'informazione del range and bearing in base al colore raggiunto, richiamando i robot nelle vicinanze. Così facendo i robot che ricevono il segnale vengono informati di dove la patch si trova e di che colore essa sia.

Nel caso la patch sia nera, il robot calcola un punto vicino al compagno per affiancarsi, mantenendo il collision avoid funzionante, in modo tale da evitare di spingere i robot sulla circonferenza all'interno. Se invece il robot è richiamato verso la patch bianca, il proprio comportamento di collision avoid viene inibito.

al fine di far avvicinare il più possibile il robot al suo aiutante. Infatti all'interno della patch bianca, il collision avoid è nuovamente attivato, facendo spostare i robot aiutanti verso l'interno nel momento in cui vengono raggiunti.

Per quanto riguarda invece la SCA, così come per l'SPC, il comportamento base è il random walk con collision avoid che è attivo nel momento in cui il robot non riceve alcun stimolo sensoriale di luce o di range and bearing. Nel caso di stimoli luminosi, il robot inibisce lo spostamento randomico a favore della fototassi in modo tale da raggiungere la zona sud dell'arena. Nel momento in cui il robot arriva abbastanza a sud, viene "bruciato" facendo così attivare il comportamento di anti-fototassi. In particolare, l'allontanamento avviene cercando di mantenere la luce perfettamente alle spalle del robot in modo tale da aumentare la probabilità di trovare il riparo. Nel momento in cui uno dei robot trova la shelter, così come nella SPC, esso inizia ad inviare la propria posizione, richiamando gli altri. Quest'ultimo comportamento è ovviamente il prioritario per tutte le architetture. Al fine di far entrare più robot possibili nella shelter, si è poi deciso di implementare un comportamento "Fai posto agli altri" (ParkME), che prevedeva il posizionamento dei primi robot salvi verso il muro del riparo, in modo tale da facilitare l'entrata di quelli che arrivavano.

4 Risultati

Nella seguente sezione vengono riportati i risultati ottenuti dopo aver effettuato ogni esperimento 30 volte (10 per ogni tipo di controller).

4.1 Surface and Perimeter Coverage

Per quanto riguarda risultati della SPC, la valutazione è inversamente proporzionale al valore dei punti.

Risulta quindi evidente dal boxplot come l'approccio ibrido risulti essere il migliore a livello di media e di valor minimo, seguito da quello a Sussunzione che nonostante un valor massimo discretamente alto rispetto agli altri due, ottiene un valor medio molto simile all'ibrido.

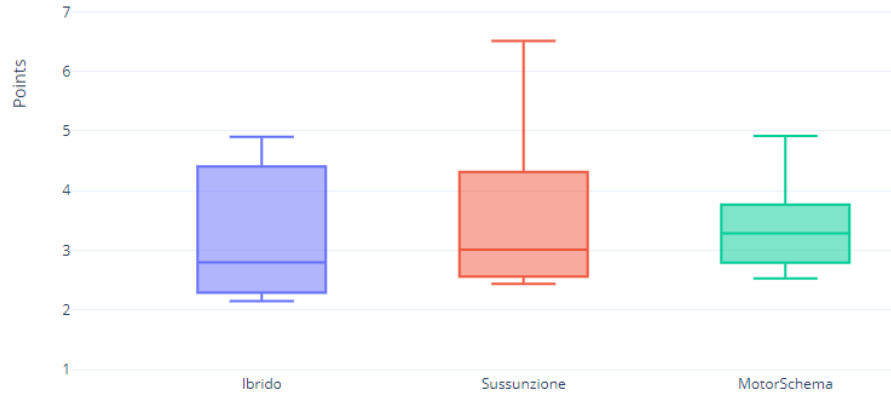


Figure 1: Boxplot risultati SPC

4.2 Shelter with Constraint Access

Contrariamente a quanto visto nella SPC, in questo esperimento la valutazione del controller è direttamente proporzionale ai punti ottenuti.

Dunque, il risultato migliore lo ottiene il modello a Sussunzione che ottiene il punteggio più alto sotto ogni aspetto (media, percentili e valori massimi e minimi). Sono paragonabili, invece, gli altri due modelli, che propongono valori molto al di sotto rispetto alla Sussunzione, con il MotorSchema che risulta essere lievemente migliore del Ibrido.

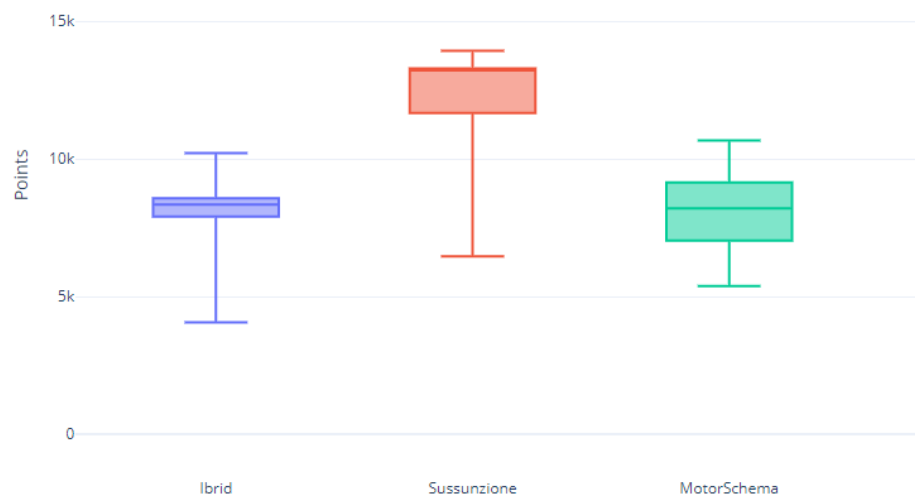


Figure 2: Boxplot risultati SCA

Referenze

- [1] Gianpiero Francesca et al. “AutoMoDe-Chocolate: automatic design of control software for robot swarms”. In: *Swarm Intelligence* 9 (June 2015). DOI: 10.1007/s11721-015-0107-9.
- [2] Carlo Pinciroli et al. *ARGoS: a Modular, Parallel, Multi-Engine Simulator for Multi-Robot Systems*. Berlin, Germany, 2012.
- [3] Allsey87. *Luabt*. URL: <https://github.com/allsey87/luabt>.